APPLICATION

FOR

UNITED STATES LETTERS PATENT

APPLICANT NAME: John J. Gregson

TITLE: METHOD AND SYSTEM TO MEASURE DISTRIBUTED SYSTEM'S RELATIVE SIZE

DOCKET NO. BLD920010006US1

INTERNATIONAL BUSINESS MACHINES CORPORATION

CERTIFICATE OF MAILING UNDER 37 CFR 1.10

I hereby certify that, on the date shown below, this correspondence is being deposited with the United States Postal Service in an envelope addressed to the Assistant Commissioner for Patents, Washington, D.C., 20231 as Express Mail Post Office to Addressee Mailing Label No. EK602285995US on 08/22/01

Bethany J. Fitzpatrick
Name of person mailing paper

Bethany J. Fitzpatrick 8/22/01
Signature Date

METHOD AND SYSTEM TO MEASURE DISTRIBUTED SYSTEM'S RELATIVE SIZE

Background Of The Invention

Field of the Invention

This invention generally relates to distributed computer systems, and more specifically, the invention relates to methods and systems to provide relative measurements of elements of distributed computer systems.

Prior Art

Vendors, consultants, and outsourcers have been searching for a method to normalize the size and function of servers, such as UNIX and Intel servers, with regard to the labor required to support them. This is a critical requirement because if one has a method to do this, then one can compare the labor to support one set of servers to another and account for the difference in relative sizes and functions of the servers.

Attempts have been made to use performance benchmarks to measure the relative size of servers e.g., SPECMarks from the SPEC organization or TPC-Cs from the TPC organization. These are impractical to use for the following reasons. The algorithms used are tied to current technology. As the technology changes, a new algorithm is published. For example, the original SPEC algorithm was targeted to uniprocessor systems. As multiprocessor system were developed, that algorithm was replaced with a newer one. Another reason that these prior art approaches are impractical is that they are published for a very small subset of servers. There is, accordingly, an important need for a method to measure all servers.

Summary Of The Invention

An object of this invention is to provide a method and system to measure the relative sizes of distributed computer systems.

Another object of the present invention is to provide a method and system to normalize the comparison of one computer server to another.

These and other objects are attained with a method and system for measuring the size of a distributed computer system of interconnected servers. A weighted asymptotic function of the count of installed CPUs on each server is computed. A CPU factor is computed based on the server architecture and operating system. A second weighted asymptotic function of the amount of RAM installed on each server is computed. A normalizing factor representing a reference date, and a RAM factor based on server architecture is computed. The product of the two weighted factors is computed for each server, and the sum of the product for each interconnected server is computed. This sum indicates a normalized measure for the size of a server. Included in the computation are factors, such as "Server Image Power Raging" and Server Image Power class."

Further benefits and advantages of the invention will become apparent from a consideration of the following detailed description, given with reference to the accompanying drawings, which specify and show preferred embodiments of the invention.

Brief Description Of The Drawings

Figure 1 outlines a method embodying an aspect of this invention.

10223033.062502

Figure 2 outlines a method embodying a second aspect of the present invention.

Figure 3 illustrates the components of a computer system that may be used to practice this invention.

Detailed Description Of The Preferred Embodiments

Generally, the present invention provides a means to normalize the comparison of one computer server to another. With reference to Figure 1, the method, generally, is to measure the relative size of the server and assign a relative weight to the size. Then to assign a relative weight based on the type of application running on the server. The product of the two is the normalized weight of the server.

More specifically, a formula is provided that allows one to determine whether a computer server is relatively equal, larger, or smaller in size to another. This formula calculates a value, Server Image Power Rating (SIPR), as follows:

SIPR = (Image CPUs) x (Image RAM GB)

Thus, in accordance with this formula, the Server Image Power Rating is equal to the product of the number of central processing units (CPUs) in the image and the amount of gigabytes of processor main memory in that image.

The Server Image Power Rating is then applied to the following table to derive a value referred to as the Server Image Power Class (SIPC)

SIPR	SIPC
≤ 1	1

$\leq 2 > 1$	2
$\leq 25 > 2$	3
$\leq 100 > 25$	4
$\leq 512 > 100$	5

From the SIPC, one assigns the relative weight. The table can be further expanded to accommodate increasingly large systems.

An alternate approach is to include the relative I/O of the server as an additional factor in the SIPR formula. The formula then becomes:

SIPR = (Image CPUs)x(Image RAM GB)x(I/O Adapter for the Image)

Thus, in accordance with this formula, the Server Image Power Rating is equal to the product of the number of central processing units (CPUs) in the image, the amount of gigabytes of processor main memory in that image, and the number of input/output cards installed in that image.

As an example, three categories can be assigned to the types of applications that run on servers: Simple, Medium and Complex. For example, a server providing file and print functions would be rated a Simple server, and a server providing database functions would be rated as a Complex server.

From this rating, the relative weight is assigned. Also, as above, the categories can be expanded to accommodate additional types of applications.

The above-described procedure may be used in a number of ways, some of which are discussed below.

Benchmarking

When a service provider delivers services in outsourcing contracts, there are times when the provider is subject to benchmarking clauses. These are onerous and place the provider at a disadvantage. The consultants to perform the benchmarking compare one environment to others and these comparisons are flawed because there is no method to normalize the relative sizes or the functions being compared. The present invention can be used to remove these discrepancies.

Financial

In the past, service providers have signed contracts which did not take computer size or function into account. This has placed providers in a position of financial risk because of changing technology. As computer servers become larger, greater skill is required to support them, which burdens providers with greater labor cost. The same is true with changes in client application mixes. Not having a method to mitigate this exposes service providers to financial loss.

Business

An important consideration to service providers is how well are they performing in IGS. In order to determine this, one of the measurements they need to understand is the labor productivity that they are achieving in their accounts. However, as technology changes and servers get larger or applications change, the service providers cannot make this comparison without a method to account for that change.

Other

In these endeavors, there are a large number of servers being supported. A significant problem is a method to count things in these environments. The above-disclosed approach simplifies the counting and provides a means by which one can estimate the relative sizes and functions with a high degree of accuracy.

The above-described approach is a significant improvement over the prior art because it accounts for the change in technology. As an example, a computer in the late 1970s might have had one CPU and 64K of RAM. A more recent computer system might have eight CPUs and 2GB RAM. From the above formula, the SIPRs for these servers would be 0.00006 and 16 respectively. The SIPCs would be 1 and 3, respectively.

In accordance with another aspect of this invention, an algorithm is provided to measure two elements of computer systems. The first is the relative power of a computer system. The second is the relative administrative effort to manage computer systems.

With reference to Figure 2, this method measures the size of a distributed system of interconnected servers. The method comprises the steps of, for each said server, forming a first weighted asymptotic function of the count of CPUs installed on said each said server, and a CPU factor based on said server architecture and operating system; and for each said server, forming a second weighted asymptotic function of the amount of random access memory installed on said each said server, and a normalizing factor representing a reference date, and a RAM factor based on said server architecture and operating system. The method comprises the further steps of forming a product of said first weighted asymptotic function and said second weighted asymptotic function for said each said server, and forming the sum of said product for all said interconnected servers.

This algorithm can be represented by the following equation:

$$F_x = (1 + w_c \bullet \log_2(c) \pm e_c) \bullet (1 + w_r \log_2(r/R_y) \pm e_r)$$

Where:

F_x is the factor be measured and is either P for the relative power measurement or A for the relative administrative effort.

w_c is a weighing factor dependent on F_x and based on the Central Processing Units (CPUs) installed on the system.

w_r is a weighing factor dependent on F_x and based on the amount of Random Access Memory (RAM, a.k.a. main memory) installed on the system.

c is the count of CPUs installed on the system.

r is the count of RAM installed on the system in units of megabytes (MB) divided by c, the count of CPUs installed on the system.

R_y is a normalization factor which represents the vase reference year RAM.

e_c is a CPU estimating factor based on the system architecture and operating system.

e_r is a RAM estimating factor based on the system architecture and operating system.

As examples, the values for the calculation of the relative power may be,

$w_c = 2$

$w_r = 0.20$

$R_y = 512$

and for the calculation of the relative administrative cost,

$w_c = 0.30$

$w_r = 0.20$

$R_y = 512$

Furthermore, the algorithm can be presented in the more general form:

$F_x = f_{CPU}(c) \cdot f_{RAM}(r)$

where f_{CPU} and f_{RAM} represent asymptotic functions of the form:

$f_x = A(x) \pm e_x$

where the accuracy of the slope of the asymptote correlates to the order of magnitude of the estimating factor e_x. For example, second and third degree polynomials, of the inverted form

$y = ax^3 + bx^2 + c$

can be used to vary the accuracy.

As will be understood by those skilled in the art, any suitable computing or calculating system or apparatus may be used to practice this invention. For example, a suitable computer system illustrated at 30 in Figure 3 may be used. System 30, generally, comprises a series of CPUs, a cache subsystem 34, and a Random Access Memory RAM) 36. Also, as will be understood by those skilled in the art, the present invention may be embodied in a computer program storage device (including software embodied on a magnetic, electrical, optical or other storage device) for normalizing the comparison of computer servers.

While it is apparent that the invention herein disclosed is well calculated to fulfill the objects stated above, it will be appreciated that numerous modifications and embodiments may be devised by those skilled in the art, and it is intended that the appended claims cover all such modifications and embodiments as fall within the true spirit and scope of the present invention.

10093743.052402